IJIREEICE

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering

Vol. 9, Issue 6, June 2021

DOI 10.17148/IJIREEICE.2021.9638

An Improved Design For A High Speed Psuedorandom Bit Generator

Ashly George[1] and Bency Varghese A.[2]

P.G Scholar, M.Tech in VLSI Design, IES College of Engineering Chittilappilly[1]

Assistant Professor, ECE, IES College of Engineering Chittilappilly [2]

Abstract: Pseudorandom bit generator (PRBG) is an essential component for securing data during transmission and storage in various cryptography applications. Among popular existing PRBG methods like linear feedback register (LFSR), linear congruential generator (LCG), coupled LCG (CLCG), and dual-coupled LCG (dual-CLCG), the latter proves to be safer. The hardware implementation of this method features a bottleneck due to the involvement of inequality equations. Initially, a direct architectural mapping of the dual-CLCG method is performed. Since two inequality equations are involved for coupling, it generates pseudorandom bit at unequal interval of your time that results in large variation in output latency. Besides, it consumes an out sized area and fails to realize the maximal period. Hence, to overcome the aforesaid drawbacks, a new efficient PRBG method, i.e., “coupled variable input LCG (CVLCG)”and its architecture came into existence. The novelty of this method is the coupling of two newly formed variable input LCGs that generates pseudorandom bit at every uniform clock rate, attains maximum length sequence and reduces one comparator area as compared to the dual-CLCG architecture. The CVLCG architecture have been modified again to enable cross coupling in order to achieve high speed and more randomness. The proposed Cross-CVLCG architecture is implemented using Verilog-HDL and evaluated for randomness using the NIST standard test tool. Experimental result reports that the method passes the randomness test with a high degree of consistency.

Keywords: Pseudorandom Bit Generator (PRBG); VLSI Architecture; CVLCG Architecture

I. INTRODUCTION

Random number generation, which is a key component in cryptographic applications, is necessary for protecting the secrecy of information from external attacks by making it unpredictable. There are two types of random number generators: Pseudorandom number generator (PRNG) and True random number generator (TRNG). Nowadays there exists the need of random numbers, not only for entertainment purposes but for computer simulations and statistical sampling, where pseudorandom number has its application. A large set of cryptographic applications like generation of secret key, generation of seed for a PRNG based system, depend on generation of pseudorandom numbers. PRNGs as it name signifies, they are not true random numbers, that is they are less unpredictable.

Security and privacy over the internet is the most sensitive and primary objective to protect data in various Internet-of-Things (IoT) applications. Millions of devices which are connected to the web generate big data which will cause user privacy issues. Also, there are significant security challenges to implement the IoT whose objectives are to attach people-to-things and things-to-things over the web. The pseudorandom bit generator (PRBG) is an important component to manage user privacy in IoT enabled resource constraint devices. A high bit-rate, cryptographically secure and enormous key size PRBG is difficult to achieve thanks to hardware limitations which demands efficient VLSI architecture in terms of randomness, area, latency and power.

The PRBG is assumed to be random if it satisfies the fifteen benchmark tests of National Institute of Standard and Technology (NIST) standard. The outputs of such generators could also be utilized in many cryptographic applications, like the generation of key material. Generators suitable to be used in cryptographic applications may have to satisfy stronger requirements than for other applications. In particular, their outputs must be unpredictable in the absence of knowledge of the inputs.

First cryptographically secure pseudorandom number generator [7], in terms of unpredictability, was introduced by Adi Shamir in 1981 who is one among the inventors of RSA [8]. Shamir uses the core concept of RSA i.e. the matter of integer factorization to make sure the safety of his new CSPRNG, with the idea of the strength of this CSPRNG is like the safety of the RSA cryptosystem due to intractability of the matter of integer factorization. On the opposite hand, since it uses modular exponentiation of giant numbers, makes some time also as not suitable for practical applications.

DOI 10.17148/IJIREEICE.2021.9638

Zenner et. al.[4] and Stern et. al.[5] reported, linear feedback register (LFSR) and linear congruential generator (LCG) are the foremost common and low complexity PRBGs. However, these PRBGs badly fail randomness tests and are insecure thanks to its linearity structure. Numerous studies on PRBG supported LFSR, chaotic map and congruent modulo are reported. Among these, Blum-Blum-Shub generator (BBS)[6] is one among the proven polynomial time unpredictable and cryptographic secure key generator due to its large prime factorize problem. Although it's secure, the hardware implementation is sort of challenging for performing the massive prime integer modulus and computing the massive special prime integer. There are various architectures of BBS PRBG, discussed in Panda et. al.[7] and Lopez et.al.[8]. Most of them either consume an outsized amount of hardware area or high clock latency. Blum Blum Shub (B.B.S.) may be a pseudorandom number generator proposed in 1986 by Lenore Blum, Manuel Blum and Michael Shub (Blum et al., 1986). Blum Blum Shub takes the form: $X_{n+1} = X\ 2^n \bmod n$ Where $n = p \times q$ is that the product of two large primes p and q. At each step of the algorithm, some output springs from X_{n+1}; the output is usually the bit parity of X_{n+1} or one or more of the least significant bits of X_{n+1}. The two primes, p and q, should both be congruent to 3 (mod 4) (this guarantees that every quadratic residue has one square root which is additionally a quadratic residue) and $\gcd(\varphi(p-1),\varphi(q-1))$ should be small (this makes the cycle length large).

A bit "1" is output if the primary LCG produces an output that's greater than the output of the second LCG and a touch "0" is output otherwise. Breaking this scheme would require one to get the seeds of the 2 independent generators, given the bits of the output bit sequence. The problem of uniquely determining the seeds for the CLCG requires
(i) a knowledge of at least $\log_2 m^2$ (m being the LCG modulus) bits of the output sequence
(ii) the solution of at least $\log_2 m^2$ inequalities where each inequality (dictated by the output bit observed) is applied over positive integers.

To mitigate the aforesaid problems of LFSR and LCG based PRNGs, a low hardware complexity coupled LCG (CLCG) has been proposed in Katti et. al[9]. The coupling of two LCGs in the CLCG method makes it more secure than a single LCG and chaotic based PRBGs that generates the pseudorandom bit at every clock cycle. Despite an improvement in the security, the CLCG method fails the discrete Fourier transform (DFT) test and five other major NIST statistical tests. DFT test finds the periodic patterns in CLCG which shows it as a weak generator. To amend this, Katti et al. [10] proposed another PRBG method, i.e. dual-CLCG that involves two inequality comparisons and four LCGs to generate pseudorandom bit sequence. The dual-CLCG method generates one-bit random output only when it holds inequality equations. Therefore, it is unable to generate pseudorandom bit at every iteration. Hence, designing an efficient architecture is a major challenge to generate random bit in uniform clock time.

II. ARCHITECTURE AND DESIGN DETAILS OF CVLCG METHOD

To overcome the aforesaid shortcomings in the dual-CLCG method and its architecture, a new PRBG method and its architecture have been designed. The coupling of two or four LCGs is the main principle for generating a pseudorandom bit sequence in the CLCG and dual-CLCG methods which make them more secure than any other linear PRBGs. Therefore, the same coupling concept is used as post-processing in the proposed PRBG method for generating pseudorandom bit at every iteration. Before post processing, a new formulation of congruential equation is used instead of using LCG. The new form of congruential generator is the class of LCG in which a constant parameter is replaced with a variable input parameter and is defined as follows,

$$x_{i+1} = a_1 \times x_i + p_i \bmod 2^n \qquad (1)$$
$$y_{i+1} = a_2 \times y_i + q_i \bmod 2^n \qquad (2)$$

The newly formed equations (1) and (2) are named as variable input linear congruential generators (Vi-LCG), where, the variable parameters and are obtained from two different LCGs and are defined as,

$$p_{i+1} = a_3 \times p_i + b_3 \bmod 2^n \qquad (3)$$
$$q_{i+1} = a_4 \times q_i + b_4 \bmod 2^n \qquad (4)$$

The pseudorandom bit sequence Z_i is obtained by using the inequality equation (5) in the post processing as follow,

$$Z_i = \{1, \text{ if } x_{i+1} > yi+1\ ; \quad 0, \text{ else} \qquad (5)$$

Where,

DOI 10.17148/IJIREEICE.2021.9638

$$B_i = \{1, \text{if } xi+1 > yi+1 \quad 0, \text{ else};$$
$$C_i = \{1, \text{if } pi+1 > qi+1 \quad 0, \text{ else}$$

Here, a_1, b_1, a_2, b_2, a_3, b_3, a_4 and b_4 are the constant parameters; x_0, y_0, p_0 and q_0 are the initial seeds. The mathematical equations (1) and (2) signify the congruential modulo 2^n addition process where the variables p_i and q_i are separately calculated from two LCGs as specified in equation (3) and (4) respectively. The pseudorandom bit is obtained by the comparison of two Vi-LCG outputs x_{i+1} with y_{i+1} as stated in equation (5) in the post processing stage. Since only one inequality condition involves in the PRBG method, it generates a pseudorandom bit at every iteration without skipping any value at the output. Moreover, it reduces one comparison process and one tristate logic as compared to the dual-CLCG method. Due to coupling of two variable-input LCGs, the PRBG method is referred as "Coupled Vi-LCG" or "CVLCG" in short. The architecture is shown in figure 1.

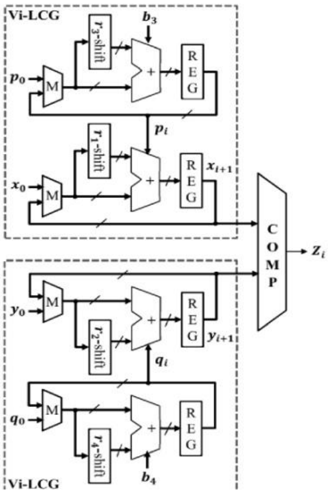

Fig 1: VLSI Architecture for CVLCG Method

III. VLSI ARCHITECTURE OF THE CROSS-CVLCG METHOD

An efficient VLSI architecture of the Cross-CVLCG method is developed to obtain the pseudorandom bit at equal interval of time. The architecture is developed by mapping the two variable input LCG (Vi-LCG) equations and one inequality equation as shown in Fig. 2. The architecture mapping of Vi-LCG is highlighted with a dotted line box. It computes n-bit binary output x_{i+1} and y_{i+1} from (x_i,y_i) and (p_i,q_i) respectively. The multiplication in the Vi-LCG and LCG equations is implemented with logical shift operation and three-operand adder when $a_i = (2^{ri}+1)$ is considered. The inequality equation (5) is realized with a binary comparator that compares the n-bit binary output x_{i+1} with y_{i+1} and produces one-bit output z_i in every equal interval of time. The architecture consumes the area of four n-bit multiplexers, four n-bit registers, four n-bit three-operand modulo 2^n adders and one n-bit comparator.

It is observed that the Cross-CVLCG architecture significantly reduces the area of one comparator when compared with the architecture designed for the dual-CLCG method. Furthermore, it generates pseudorandom bit at every uniform clock cycle, whereas the architecture developed for the dual-CLCG method is unable to generate pseudorandom bit at every uniform clock rate.

The coupling of two or four LCGs is the main principle for generating a pseudorandom bit sequence in the CLCG and dual-CLCG methods which make them more secure than any other linear PRBGs. Therefore, the same coupling concept is used as post-processing in the proposed PRBG method for generating pseudorandom bit at every iteration. Before post processing, a new formulation of congruential equation is used instead of using LCG. The new form of congruential generator is the class of LCG in which a constant parameter is replaced with a variable input parameter.

Linear congruential generators (LCGs) of the form $x_{i+1} = a\, x_i + b \pmod{m}$, have been used to generate pseudorandom numbers. However these generators have been known to be insecure. This implies that if a small sequence of numbers generated by an LCG is known then it is possible to predict the remaining numbers in the sequence that will be generated. Thus it is propose to generate a secure pseudorandom bit sequence by cross coupling two LCGs as follows.

A 1 is output if the first LCG produces an output that is greater than the output of the second LCG and a 0 is output otherwise.

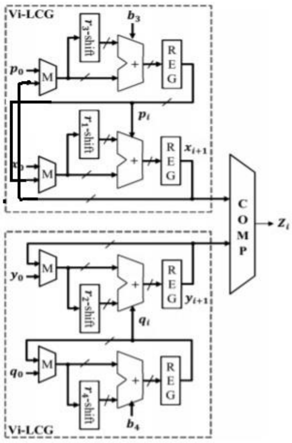

Fig 2: VLSI Architecture for Cross-CVLCG Method-1

The proposed architecture consumes the area of four n-bit multiplexers, four n-bit registers, four n-bit three-operand modulo 2^n adders and one n-bit comparator. The output of the second LCG is coupled to the input of the first one and vice versa, hence cross coupling is done. Before post processing, a new formulation of congruential equation is used instead of using LCG. The new form of congruential generator is the class of LCG in which a constant parameter is replaced with a variable input parameter.

Also the architecture can be modified to get more randomness by coupling the controlled LCG part too. That means the output of the fourth LCG can be coupled to the input of the third LCG.

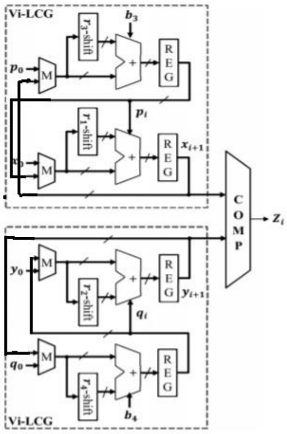

Fig 3: VLSI Architecture for Cross-CVLCG Method-2

In both of the cross-coupled cases the system will be tend to more pseudorandom as it will be having some of the behaviour of a flip flop and will be generating more random sequences.

IV. IMPLEMENTATION RESULTS

The CVLCG method and Cross-CVLCG methods are simulated on Xilinx Vivado Software tool on Virtex5, Virtex7 and Zynq-7000 FPGA devices. This section presents the physical implementation results of the CVLCG method and its cross-coupled modifications of 16-bit word size. Further, the statistical test for randomness is performed on the captured random bit sequences by using the MATLAB tool for measuring randomness. The CVLCG design and the proposed Cross - CVLCG design of 16- bit word size is implemented on Xilinx Virtex5, Virtex7 and Zynq-7000 FPGA devices to ensure a fair comparison. The performance on Virtex7 and Zynq-7000 FPGA devices reports the same results due to both fabricated with same 28nm process technology. The proposed Cross - CVLCG architecture is compared with the existing CVLCG PRBG methods considered for the timing comparison in Table 1.

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

Vol. 9, Issue 6, June 2021

DOI 10.17148/IJIREEICE.2021.9638

Table 1 : Delay Comparison Between CVLCG and Cross-CVLCG Methods

Method	Critical Path Delay
CVLCG Method	7.478 ns
Cross-CVLCG Method-1	7.002 ns
Cross-CVLCG Method-2	6.217 ns

The FPGA implementation results show that the CVLCG architecture is indeed capable of generating pseudorandom bit with the latency of one clock cycle and less hardware resources. The battery of randomness runs test results report that it successfully passed with a high degree of consistency. The advantages such as less area, low latency, low power, long period, pseudorandom bit generation at the uniform clock rate and a high degree of randomness in the proposed Cross - CVLCG method makes it suitable for real-time secure information exchange and data security in the light weight IoT enabled smart devices.

The device utilization of all the resources used in the design in all the three methods are enlisted in the below table.

Table 2: Device Utilization Summary of CVLCG Method

Resource	Utilization	Available	% of Utilization
Slice LUTs	567.0	53200.0	1
Slice Registers	714.0	106400.0	1

The randomness properties of binary sequences can be evaluated using a battery of statistical tests provided by the standard NIST 800-22 [12]. The tests statistically determine if a given binary sequence is consistent with the output of a random sequence, the null hypothesis being that the sequence is random. The NIST Test Suite, consisting of 15 tests, is a statistical package that tests the randomness of (arbitrarily long) binary sequences. These sequences can be produced either by hardware or by software based random number generators.

MATLAB can be used to test whether a sequence is random or not. Run test can be performed to check whether the sequence generated are random or not. The test is based on the number of runs of consecutive values above or below the mean of x. The result h is 1 if the test rejects the null hypothesis at the 5% significance level, or 0 otherwise.

The statistical test for randomness, i.e, The Runs Test for the existing CVLCG method and the proposed Cross-CVLCG methods are done on MATLAB. The returned value of h = 0 indicates that runstest does not reject the null hypothesis that the values in the array x, which contains the generated pseudorandom numbers are in random order.

V. CONCLUSION

The aim of this work is to develop an efficient hardware architecture for generating pseudorandom bit sequence by targeting the IoT enabled applications. Initially, the dual-CLCG algorithm is studied to perform its hardware architecture. However, it experiences the drawbacks of generating pseudorandom bit at an unequal interval of time that lead to varying in the output latency and also fails to achieve the maximum length period. Therefore, the aforesaid drawbacks are overcome by proposing a new efficient PRBG method, i.e. “coupled variable-input LCG (CVLCG)” and its VLSI architecture. The CVLCG method is developed by coupling of two variable-input LCGs, which generates pseudorandom bit at every iteration and also attains the maximum length sequence. Moreover, it saves one comparator area in comparison to the dual-CLCG architecture. Indeed the CVLCG architecture is modified by enabling the cross-coupling in order to achieve more randomness and high speed. The cross-coupling is done on the controller CVLCG part and as well as controlled CVLCG part.

The behavioural results show that the proposed Cross-CVLCG architectures are indeed capable of generating pseudorandom bit with the latency of one clock cycle and less hardware resources. The critical path delay for the existing CVLCG method and the proposed Cross-CVLCG methods have been analysed and the delay for the existing CVLCG method is estimated to be 7.478ns and that of the Cross-CVLCG method-1 has a time delay of 7.002ns. Among the three methods implemented the Cross-CVLCG method-2 is found to be more faster and the critical path delay is estimated to be 6.217ns. The device utilization summary had shows that the number of LUTs are less compared to the previous PRBG methods and thus having less area. The randomness analysis has done using the Runs Test and the result shows that the three implemented methods are random and among the three methods Cross-CVLCG method-2 is found to having more randomness.

The advantages such as high speed, less area, low latency, long period, pseudorandom bit generation at the uniform clock rate and a high degree of randomness in the proposed Cross-CVLCG methods makes it suitable for real-time secure information exchange and data security in the lightweight IoT enabled smart devices and in other cryptographic applications.

REFERENCES

[1] A.K. Panda and C.K. Ray "A Coupled Variable Input LCG Method and its VLSI Architecture for Pseudorandom Bit Generation., " IEEE Transactions on Instrumentation and Measurement, January 2020

[2] A.K. Panda and C.K. Ray "Modified Dual-CLCG Method and Its VLSI Architecture for Pseudorandom Bit Generation" IEEE Transactions On Circuits And Systems–I: Regular Papers January 2020

[3] E. Fernandes, A. Rahmati, K. Eykholt and A. Prakash, "Internet of things security research: A rehash of old ideas or new intellectual challenges," IEEE Security & Privacy, vol. 15, no. 4, pp. 79-84, 2017.

[4] F. Firouzi, B. Farahani, M. Ibrahim and K. Chakrabarty, "From EDA to IoT eHealth: promise, challenges, and solutions," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. PP, no. 99, pp. 1-1, 2018.

[5] G. Mois, S. Folea and T. Sanislav, "Analysis of three IoT-based wireless Sensors for environmental monitoring," in IEEE Transactions on Instrumentation and Meas., vol. 66, no. 8, pp. 2056-2064, Aug. 2017.

[6] K. Yu, M. Arifuzzaman, Z. Wen, D. Zhang and T. Sato, "A key management scheme for secure communications of information centric advanced metering infrastructure in smart grid,", in IEEE Trans. on Instrumentation and Meas., vol. 64, no. 8, pp. 2072-2085, Aug. 2015.

[7] O. Goldreich, “Foundations of Cryptography II: Basic Applications." Cambridge University Press, 2009, New York, USA.

[8] J.G.Pandey,T.GoelandA.Karmakar, "A High-Performance and Area-EfficientVLSI Architecture for the PRESENT Light weight Cipher," 31st International Conference on VLSI Design and 17th Int. Conference on Embedded Systems (VLSID), Pune, 2018, pp. 392-397.

[9] D. E. Knuth, “Deciphering a linear congruential encryption,” in IEEE Trans. Inform. Theory, vol. 31, no. 1, pp. 49–52, Jan 1985.

[10] E. Pasalic, "On Guess and Determine Cryptanalysis of LFSR-Based Stream Ciphers," in IEEE Transactions on Information Theory, vol. 55, no. 7, pp. 3398-3406, July 2009.

[11] M. Matsumoto, T. Nishimura, “Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator,” ACM Trans. Model. Comput. Simul, vol. 8, no. 1, pp. 3-30, 1998.

[12] G.Marsaglia and L.-H.Tsay, “Matrices and the structure of random numbers equences,” Linear Algebra Appl., vol. 67, pp. 147–156, 1985.

[13] A. B. Orúe López, L. Hernández Encinas, A. Martín Muñoz and F. Montoya Vitini, “A lightweight pseudorandom number generator for securing the internet of things,” in IEEE Access, vol. 5, pp. 2780027806, 2017.

[14] J. Stern, “Secret linear congruential generators are not cryptographically secure,” in Proc. 28th Annu. Symp. Found. Comput. Sci., Oct. 1987, pp. 421–426,

[15] Mohammed Bakiri, Christophe Guyeux, Jean-François Couchot, Abdelkrim Kamel Oudjida, “Survey on hardware implementation of random number generators on FPGA: Theory and experimental analyses,” Computer Science Review, vol. 27, pp. 135-153, 2018.

[16] G. Bhatnagar and Q. M. J. Wu, "Chaos-based security solution for fingerprint data during communication and transmission," IEEE Trans. on Instrumentation and Meas., vol. 61, no. 4, pp. 876-887, April 2012.

[17] M. Bakiri, C. Guyeux, J. F. Couchot, L. Marangio and S. Galatolo, “A hardware and secure pseudorandom generator for constrained devices,” in IEEE Trans. on Industrial Informatics, vol. 14, no. 8, pp. 3754-3765, Aug. 2018.

[18] M.Garcia-Bosque,A.Pérez-Resa,C.Sánchez-Azqueta,C.AldeaandS.Celma, "ChaosBased Bitwise Dynamical Pseudorandom Number Generator On FPGA," in IEEE Transactions on Instrumentation and Measurement, vol. 68, no. 1, pp. 291-293, Jan. 2019

[19] L. De la Fraga, E. Torres-Pérez, E. Tlelo-Cuautle, and C. Mancillas López, “Hardware implementation of pseudo-random number generators based on chaotic maps,” Nonlinear Dynamics, Springer, vol. 90, pp. 1661–1670, Nov. 2017.

[20] L. Blum, M. Blum, and M. Shub, SIAM J. “A simple unpredictable pseudorandom number generator,” Comput., vol. 15, no. 2, pp. 364–383, 1986